

# Programación Web

[i92mogra@uco.es](mailto:i92mogra@uco.es) | Antonio Moruno Gracia

[i92pedud@uco.es](mailto:i92pedud@uco.es) | David Pérez Dueñas

[i92rigam@uco.es](mailto:i92rigam@uco.es) | Marcos Rivera Gavilán

## Práctica 1

10 de octubre del 2021

### Visión general

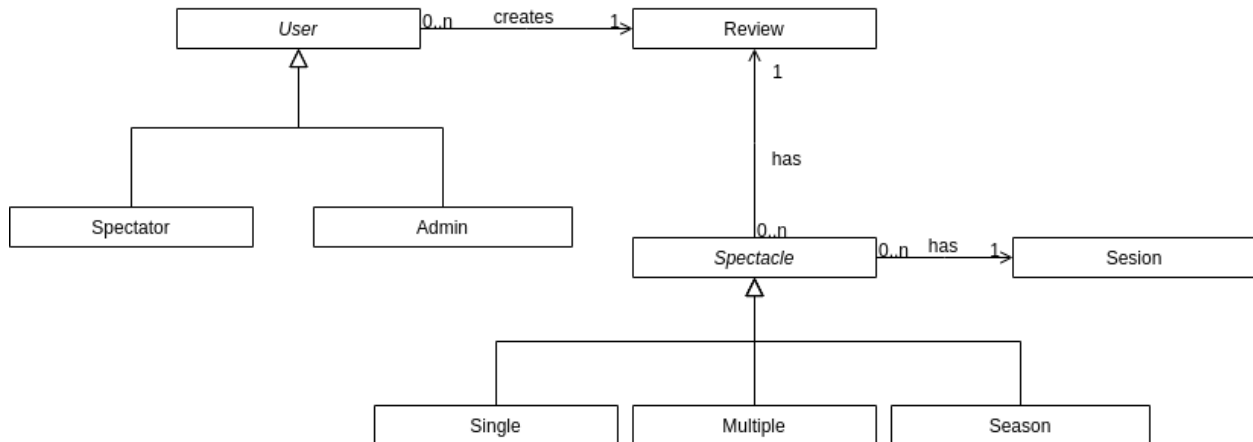
La idea del grupo a la hora de desarrollar esta práctica ha sido la de intentar crear un programa que cumpliese con los requisitos del sistema especificados en el guión de la práctica y que además pudiese ser fácilmente modificable y escalable pues su esqueleto servirá de base para el resto de prácticas.

Algunos detalles sobre el sistema a tener en cuenta son:

1. **Interfaz:** Pese a que la interfaz del programa fuese la consola hemos intentado que sea lo más accesible posible y que resulte intuitiva para todos los usuarios.
2. **Almacenamiento:** El almacenamiento del sistema se realiza en ficheros de texto plano donde los datos se encuentran separados por comas. Los ficheros se cargan en memoria al principio del programa y se sobrescriben cuando se acaba la ejecución
3. **Diseño:** Respecto al diseño del programa hemos intentado desacoplar lo máximo posible la parte que interactúa con el usuario con entrada y salida de la parte que se encarga de controlar la lógica del sistema.

## Funcionamiento

Para explicar mejor el funcionamiento del sistema usaremos el siguiente diagrama de clases:



Tal y como se puede observar en el diagrama, las clases Espectador y Administrador heredan de usuario y los distintos tipos de espectáculos heredan de un espectáculo general. Además disponemos de una clase Review y otra sesión que serán la que interaccionarán con los espectáculos y los usuarios.

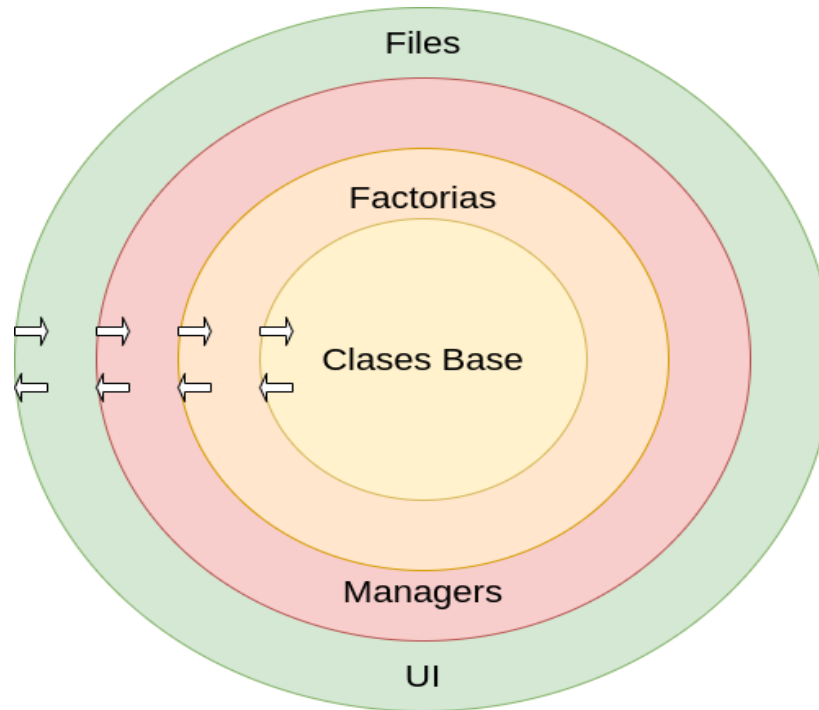
Por encima de esta primera capa se encuentran las clases que implementan el patrón Factory y dan lugar a las factorías de usuarios y de espectáculos que serán las encargadas de instanciar los distintos tipos de objetos de estas clases. Para ello harán uso del método estático `getInstance()` que nos devolverá en cada caso un usuario o un espectáculo del tipo indicado. (Esto es posible gracias al Principio de Sustitución de Liskov).

Si seguimos avanzando en el modelado de la aplicación nos encontramos con los managers. Cada tipo de clase base (Usuario, Espectáculo, Reseña y Sesión) cuentan con su propio manager implementado bajo el patrón de diseño singleton y que actúa como interfaz entre el programa y las clases.

En el último nivel se encuentran las clases utilidades y storage que contienen los distintos métodos relacionados con la interacción con el usuario y el almacenamiento de la información.

## Funcionamiento

En este diagrama se representa la interacción entre las distintas capas del sistema:



## Dificultades

Las principales dificultades a la hora de implementar la aplicación han sido dos:

### 1. Falta de experiencia

Ya que era la primera vez que nos enfrentábamos a un proyecto de este estilo con el añadido de no haber usado java nunca por lo que debíamos de consultar constantemente la documentación.

### 2. Falta de tiempo

Pese a que nos hemos organizado bien, con un poco más de tiempo podríamos haber hecho test y pulido un poco más la interfaz.

## Referencias:

Estos son los enlaces a las distintas fuentes de información que hemos empleado para realizar el proyecto.

- <https://www.javatpoint.com/java-get-current-date>
- <https://stackoverflow.com/questions/907170/java-getminutes-and-gethours>
- <https://www.baeldung.com/find-list-element-java>
- <https://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html#param>
- <https://www.javatpoint.com/how-to-clear-screen-in-java>
- <http://puntocomnoesunlenguaje.blogspot.com/2012/08/java-scanner.html>
- <https://jarroba.com/arraylist-en-java-ejemplos/>
- [https://www.w3schools.com/java/java\\_files\\_create.asp](https://www.w3schools.com/java/java_files_create.asp)
- <https://www.baeldung.com/java-csv-file-array>
- <https://www.baeldung.com/reading-file-in-java>
- <https://stackoverflow.com/questions/30832101/buffered-reader-read-text-until-character>
- [https://www.w3schools.com/java/java\\_files\\_read.asp](https://www.w3schools.com/java/java_files_read.asp)
- <https://stackoverflow.com/questions/30832101/buffered-reader-read-text-until-character>
- <https://www.javatpoint.com/how-to-take-string-input-in-java>
- <https://www.javatpoint.com/how-to-take-string-input-in-java>
- <https://www.baeldung.com/find-list-element-java>
- <https://stackoverflow.com/questions/428918/how-can-i-increment-a-date-by-one-day-in-java>
- <https://stackoverflow.com/questions/2592501/how-to-compare-dates-in-java>
- <https://refactoring.guru/es/design-patterns/factory-method>